

Amendments to the Claims:

The listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

Claim 1. (Currently Amended) A software generation system comprising:

a specification analysis means which analyzes an object-oriented specification input by a system operator, for deriving specification information;

an optimized information input means, ~~which is operable after a program is written~~, for inputting optimized information from an external unit, indicating a necessity or lack of necessity for use of a dynamic generation function for dynamic generation of an instance representing one of a set of object oriented functions;

a function removing means which checks said specification information derived by said specification analysis means and the optimized information input via said optimized information inputting means by collating with a predetermined function removal rule, which removes a function which

becomes unnecessary from a set of object-oriented functions by which members are realized, for generating from the specification information and the optimized information, program information excluding the unnecessary function; and

a code generation means for generating a code according to said program information obtained by said function removing means; [[.]]

wherein said optimized information input means includes selection means operable after said system operator has input said object oriented specification and before generation of said code, for receiving from said system operator a selection from among use and not-to-use options for said dynamic generation function.

Claims 2-7. (Cancelled).

Claim 8. (Previously Presented) The software generation system according to Claim 1, further comprising:

an analysis result display means for displaying a status of use of an object-oriented function by which a member is realized from the specification information.

Claims 9-10. (Cancelled)

Claim 11. (Currently Amended) A software generation system comprising:

a specification analysis means which analyzes an object-oriented specification for deriving specification information;

an optimization instruction input means, which is operable after a program is written, for a system operator inputting an optimization instruction via an external unit, indicating use or ~~lack of use~~ not-to-use of respective object oriented functions contained within a set of object oriented functions;

a function removing means which checks said specification information derived by the specification analysis means and said optimization instruction input via the optimization instruction inputting means, by collating with a predetermined function removal rule, which removes a function that becomes unnecessary from said set of object-oriented functions by which members are realized, for generating from the specification information and the optimization instruction entered by said system operator, program information excluding the unnecessary function; and

a code generation means for generating a code according to said program information obtained by said function removing means; [[.]]

wherein said optimization instruction input means comprises selection means, operable after said system operator has input said object-oriented specification and before generation of said code, for offering to said system operator an option to select from use and not-to-use instructions for said respective object-oriented functions.

Claim 12. (Previously Presented) The software generation system according to Claim 11, further comprising:

an analysis result display means for displaying a status of use of an object-oriented function by which a member is realized from the specification information.

Claim 13. (New) The software generation system according to Claim 1, wherein said selection means comprises display means for offering to said system operator use and not-to-use options for said dynamic generation function, which options can be selected by said system operator via an input interface.

Claim 14. (New) The software generation system according to Claim 8, wherein said selection means comprises display means for offering to said system

operator use and not-to-use options for said dynamic generation function, which options can be selected by said system operator via an input interface.

Claim 15. (New) The software generation system according to Claim 11, wherein said selection means comprises display means for displaying use and not-to-use indicators, for individual ones of said respective object-oriented functions, which indicators can be selected by said system operator via an input interface.

Claim 16. (New) The software generation system according to Claim 12, wherein said selection means comprises display means for displaying use and not-to-use indicators, for individual ones of said respective object-oriented functions, which indicators can be selected by said system operator via an input interface.

Claim 17. (New) A method for automatically generating software, using a software generation system having a processing system that includes a system operator interface, a specification analysis unit which analyzes an object oriented specification input by the system operator for deriving specification information, and a code generating unit for generating a code according to said specification information, said method comprising:

said system operator entering an object oriented specification into said processing system via said interface;

after entry of said object oriented specification, said processing system displaying to said system operator a function item selection screen, which offers to said system operator an option to select from use and not-to-use instructions for respective object oriented functions contained in a set of object oriented functions;

for each of said respective object oriented functions, said system operator selecting one of said use and not-to-use instructions;

said processing system checking said specific information derived by said specification analysis unit and said use and not-to-use instructions selected by said system operator, by collating with a predetermined function removal rule, which removes a function that becomes unnecessary from said set of object-oriented functions by which members are realized, for generating from said specification information and said instructions, program information excluding unnecessary functions; and

said code generating unit generating a code according to said program information excluding unnecessary functions.